อาจหาญ สัตยารักษ์

SOFT MAIL

เทอร์โบเบสิก
กับการเคลื่อนที่ของวัตถุ

เรียนรู้วิธีการสร้างกราฟิกขนาดปกติและขนาดใหญ่ ทั้งไทยและอังกฤษ พร้อมด้วยกฎการเคลื่อนที่

การเขียนโปรแกรมกราฟิกด้วยภาษาเบสิกในยุคก่อนนั้น ดูจะเป็นเรื่องยุ่งยากสำหรับผู้ที่มีจอโมโนโครมธรรมดา เพราะว่าไม่สามารถรันโปรแกรมที่เขียนได้ทันที จะต้องคอมพายล์ให้เป็นไฟล์ EXE เสียก่อน จากนั้นออกจากเอดิเตอร์ไปยังดอส รันไฟล์ที่จัดการให้จอโมโนโครมเป็นจอ CGA แล้วจึงรันไฟล์ EXE นั้น ๆ บนจอภาพได้

ผู้เขียนเคยเสนอแนวการเขียนกราฟิกบนเทอร์โบเบสิกในหนังสือเทคนิคการเขียนกราฟิกบนเทอร์โบเบสิกในส่วนของการรันบนจอภาพขนาด 300×200 จุดไปแล้ว ทั้งยังแสดงแนวทางการสร้างภาษาไทยเพื่อใช้สื่อความหมายในโปรแกรมภาษาเบสิกไว้แล้วด้วย

เนื่องจากเทคโนโลยีทางด้านฮาร์ดแวร์ของระบบคอมพิวเตอร์ได้เปลี่ยนไปจากเดิมหลายอย่าง ในปัจจุบันนี้เรามีจอสีและจอวีจีเอโมโนโครมที่สามารถเขียนกราฟิกในขนาดที่โตขึ้น ทำให้ความละเอียดของภาพสามารถทำได้ถึง 640×350 จุดเป็นอย่างน้อย ดังนั้นการเขียนกราฟิกบนเทอร์โบเบสิกก็ควรที่จะพัฒนาให้สร้างภาพที่ละเอียดตามไปด้วย

ก่อนที่จะเขียน CAI เรื่อง การเคลื่อนที่ของวัตถุเมื่อถูกแรงกระทำนั้น ผู้เขียนขอเสนอแนวทางใหม่ในการเขียนชุดคำสั่งแสดงอักขระภาษาไทยและภาษาอังกฤษก่อน ดังนี้

1. แนวการสร้างภาษาไทยและภาษาอังกฤษกราฟิกขนาดปกติด้วยตัวอักขระที่สวยงาม โดยการใช้คำสั่งลากเส้นตรงคือ LINE ที่มีการกำหนดรูปแบบของเส้นตรง โดยอาศัยไฟล์ข้อมูลชื่อ thaieng.fon

2. การสร้างชุดคำสั่งในการป้อนข้อความ โดยอาศัยการทำภาษาไทยในข้อ 1 มาช่วยในการแสดงผล

3. แนวการสร้างภาษาไทยและภาษาอังกฤษกราฟิกขนาดใหญ่ด้วยข้อมูลของตัวอักขระที่สวยงามจากไฟล์ที่ใช้ในการพิมพ์ลงกระดาษของโปรแกรมพิมพ์เอกสารราชวิถีเวิร์ดพีซีคือ ไฟล์ font1.lq

ในซอฟต์เมล์ประกอบบทความนี้มีไฟล์ thai.inc, thai.bas, force.bas, thaieng.fon และ font1.lq รวมทั้งสิ้น 5 ไฟล์

สร้างอักขระกราฟิกขนาดปกติ

ขอเริ่มที่การเขียนชุดคำสั่งการแสดงข้อความภาษาไทยและภาษาอังกฤษแบบกราฟิกในขนาดปกติก่อน โดยผู้เขียนจะใช้วิธีการอ่านข้อมูลจากไฟล์ข้อมูลชื่อ thaieng.fon ซึ่งมีรายละเอียดดังนี้

1. ไฟล์มีความยาว 4,080 ไบต์

2. อักษรหนึ่งตัวจะใช้ข้อมูล 16 ไบต์ ดังนั้นจึงมีจำนวนอักขระทั้งสิ้น 4080/16=255 ตัวอักษร

3. ลำดับรหัสแอสกีของอักษรจึงเริ่มที่ 1 ไปถึงรหัสแอสกี 255 (รหัสเกษตร)

4. รหัสแอสกี 1 ข้อมูลอยู่ที่ลำดับที่ 1-16, รหัสแอสกี 2 ข้อมูลอยู่ที่ลำดับที่ 17-32 ไปเรื่อย ๆ โดยคำนวณตำแหน่งเริ่มต้นได้จาก (แอสกี-1)*16+1 เช่น รหัสแอสกี 3 จะได้ (3-1)*16+1=32+1=33

โดยการลากเส้นตรงจำนวน 16 เส้นเรียงจากบนลงล่างตามข้อมูลที่ได้ก็สามารถแสดงตัวอักษรนั้น ๆ ให้ปรากฏบนจอภาพได้ ซึ่งนำไปเขียนโปรแกรมบนเทอร์โบเบสิกด้วยชุดคำสั่งชื่อ OutThaiXY ดังแสดงในรูปที่ 1

ก่อนที่จะอธิบายรายละเอียดของโปรแกรมนั้น ผู้เขียนขอทำความเข้าใจในลิสติงที่พิมพ์เสนอทั้งหมดก่อน เพราะว่าผู้เขียนต้องการให้ส่วนของโปรแกรมในบางบรรทัดไม่ยาวจนเกินไป ดังนั้นจึงทำให้บางครั้งอาจมีการเขียนคำสั่งมากบรรทัดไปบ้าง เช่น

```
IF Right $ (Conten$,1) = Chr$(215)then
  LINE(XX-7,Y-18)-(XX,Y+18,0,BF,&HFFFF
END IF
```

```
 1:  SUB OutThaiXY(X,Y,Content$)
 2:   IF ReadFile<>27 then
 3:     OPEN "ThaiEng.FON" AS #1 LEN=4080
 4:     FIELD #1, 4080 AS Font$
 5:     GET #1,1
 6:     CLOSE #1
 7:     ThaiNormal$=Font$ : Ccolor=3
 8:   END IF
 9:    ReadFile=27
10:    for k=1 to len(Content$)
11:      Ypix=0  : Xpix=0
12:    IF Mid$(Content$,k,1)=chr$(3) then
13:          k=k+1
14:       IF Mid$(Content$,k,1)>chr$(47) and Mid$(Content$,k,1)<chr$(58) then
15:           Ccolor=Asc(Mid$(Content$,k,1))-48 : k=k+1
16:           if Mid$(Content$,k,1)=chr$(1) then Ccolor=Ccolor+9 : k=k+1
17:       END IF
18:    END IF
19:    IF Mid$(Content$,k,1)>chr$(32) then
20:       if Mid$(Content$,k,1)>chr$(214) then Ypix=8 : Xpix=-8
21:         IF Mid$(Content$,k,1)>chr$(216) then
22:             Ypix=-14 : Xpix=-8
23:             if Mid$(Content$,k-1,1)>chr$(216) then Ypix=-18
24:             if Mid$(Content$,k+1,1)=chr$(207) then Ypix=-19
25:         END IF
26:         IF Mid$(Content$,k,1)=chr$(207) then
27:             circle(X-4,Y),1,Ccolor
28:             Mid$(Content$,k,1)=chr$(206)
29:         END IF
30:         ThaiPatt$=Mid$(ThaiNormal$,(ASC(Mid$(Content$,k,1))-1)*16+1,16)
31:        for No=1 to 16
32:          Patt=ASC(Mid$(ThaiPatt$,No,1))*128
33:          line(X+Xpix,Y+No+Ypix)-(X+Xpix+15,Y+No+Ypix),Ccolor,,Patt
34:       next
35:    END IF
36:      if Mid$(Content$,k,1)<chr$(215) then X=X+8
37:   next
38:  END SUB
```

รูปที่ 1 ชุดคำสั่งแสดงข้อความภาษาไทยและภาษาอังกฤษกราฟิกขนาดปกติจากไฟล์ thaieng.fon

ซึ่งทั้งสามบรรทัดนี้สามารถเขียนในบรรทัดเดียวกันได้ โดยการนำบรรทัดที่สองมาต่อกับบรรทัดที่หนึ่ง และตัด END IF ในบรรทัดที่สามทิ้งไป เป็นต้น

ผู้อ่านจะลองแก้ไขในบางส่วนของโปรแกรมที่มีในบทความนี้ใหม่ก็ได้

รายละเอียดของชุดคำสั่ง OutThaiXY มีดังนี้

1. บรรทัดที่ 2-8 เป็นการอ่านไฟล์ข้อมูลของอักขระจากไฟล์ชื่อ thaieng.fon ซึ่งจะอ่านครั้งเดียวเมื่อเรียกใช้ชุดคำสั่งนี้เท่านั้น โดยเปิดไฟล์ที่มีระเบียนข้อมูลของข้อความยาว 4,080 ไบต์ โปรแกรมจะอ่านมาเพียงหนึ่งระเบียนข้อมูล นำข้อมูลที่อ่านได้กำหนดไว้ในตัวแปรชื่อ ThaiNormal$ เพียงตัวแปรเดียว การที่ตัวแปร ThaiNormal$ สามารถเก็บข้อมูลขนาด 4,080 ไบต์ได้นั้น เป็นเพราะว่าเทอร์โบเบสิกกำหนดความยาวของตัวแปรข้อความได้มากถึง 32,767 ไบต์ ตัวแปร Ccolor คือสีเริ่มต้นของตัวอักขระ ซึ่งจะเริ่มที่สีหมายเลข 3 เมื่อเรียกใช้ครั้งแรก แต่ครั้งต่อไปตัวแปร Ccolar จะเป็น

หมายเลขที่กำหนดไว้ก่อนหน้านี้

2. บรรทัดที่ 9 กำหนดตัวแปร ReadFile=27 เมื่อยกเลิกการอ่านไฟล์ครั้งต่อไป

3. บรรทัดที่ 10-37 เป็นการแสดงข้อความที่กำหนดบนจอภาพกราฟิก

4. บรรทัดที่ 11 กำหนดตัวแปร Ypix, Xpix เพื่อใช้ในการแสดงสระและวรรณยุกต์ให้ถูกต้องตามหลักการเขียนภาษาไทย

5. บรรทัดที่ 12-18 เป็นการตรวจสอบการแสดงสีของอักขระ โดยใช้ ^C เข้าสู่ส่วนนี้ ตามด้วยตัวเลข 0 ถึง 9 (เพียงหนึ่งตัวเลข) ซึ่งจะทำให้ได้สีหมายเลข 0-9 ตามลำดับ ถ้าตามตัวเลขด้วย A จะเพิ่มค่าตัวแปร Ccolor อีก 9 ทำให้ได้สีหมายเลข 10-15 ได้

6. บรรทัดที่ 19-35 ส่วนการแสดงอักขระรหัสแอสกี 33 ขึ้นไป

7. บรรทัดที่ 20 ถ้าเป็นรหัสแอสกี 215 ขึ้นไป ให้เปลี่ยนค่าตัวแปร Ypix, Xpix เพื่อให้แสดงอักษรให้ตรงตำแหน่งของสระและวรรณยุกต์

8. บรรทัดที่ 21-25 ตรวจสระบนและวรรณยุกต์เพื่อกำหนดตัวแปร Ypix, Xpix ให้ตรงกับตำแหน่งที่แสดง โดยจะตรวจย้อนหลังมาหนึ่งอักขระ (k-1) ว่าเป็นสระบนหรือไม่ และตรวจสอบไปข้างหน้าหนึ่งอักขระ (k+1) ว่าเป็นสระ ำ หรือไม่ ทั้งนี้เพื่อให้วรรณยุกต์แสดงซ้อนบนสระบนได้อย่างสวยงาม

9. บรรทัดที่ 26-29 ตรวจการแสดงสระ ำ โดยวาดวงกลมบนอักขระก่อนหน้า และให้แสดงสระ า แทนสระ ำ ทั้งนี้เพื่อให้เกิดความสวยงาม

10. บรรทัดที่ 30 ส่วนการคำนวณหาข้อมูลของอักขระ โดยตัดข้อความมาจากตัวแปร ThaiNormal$ มาเพียง 16 ไบต์ เพื่อใช้กำหนดรูปแบบของเส้นตรง 16 แบบ

11. บรรทัดที่ 31-34 เป็นการนำข้อมูลจากตัวแปร ThaiPatt$ มากำหนดลากเส้นตรง 16 เส้น ซึ่งทำให้ได้ภาพของอักขระนั้น ๆ บนจอภาพ การที่ต้องใช้ 128 คูณกับข้อมูลที่อ่านมาก็เพราะต้องการเลื่อนบิตของข้อมูลให้ขยับไปทางซ้ายอีก 7 บิต ทำให้ภาพของแต่ละตัวอักษรมีช่องว่างด้านซ้ายหนึ่งจุด การใช้คำสั่ง line เพื่อวาดเส้นตรงขนาด 15 จุด แต่จะเกิดภาพจริง ๆ เพียง 7 จุด เพราะข้อมูลด้านขวาจะมีค่าประจำบิตเป็น 0 สีของเส้นตรงเป็นไปตามตัวแปร Ccolor และรูปแบบของเส้นจะเป็นไปตามตัวแปร Patt

โปรดสังเกตค่าตัวแปร Xpix, Ypix, No จะช่วยให้การแสดงภาพของแต่ละตัวอักษรอยู่ในตำแหน่งที่ถูกต้องตามหลักภาษาไทย

12. บรรทัดที่ 36 เมื่อพบว่าเป็นรหัสแอสกีน้อยกว่า 215 จะเลื่อนค่า x ไปทางขวาอีก 8 จุด ซึ่งทำให้อักษรตัวถัดมาแสดงต่อไปทางขวา

การป้อนข้อความ

เพื่อให้การทำงานในกราฟิกมีความสมบูรณ์ เราจึงเขียนชุดคำสั่งป้อนข้อความขึ้นมาใช้งาน โดยมีหลักการดังนี้

1. กำหนดตัวแปรข้อความ ThaiChr$ เพื่อใช้แปลรหัสแอสกีภาษาอังกฤษที่ป้อนผ่านแป้นพิมพ์ให้มาเป็นรหัสแอสกีภาษาไทย (ในกรณีที่เลือกป้อนเป็นภาษาไทย) เพราะจะสะดวกกว่าการกำหนดเป็นตัวแปรมิติ

2. กำหนดตัวแปร ThaiMode เพื่อใช้ตรวจสอบว่าป้อนภาษาไทย (ThaiMode=1) หรือภาษาอังกฤษ (ThaiMode=0)

3. แสดงเคอร์เซอร์รูปสี่เหลี่ยมในขนาดที่ต่างกันเมื่ออยู่ในภาวะการป้อนภาษาไทยหรือภาษาอังกฤษ

4. ตรวจการกดแป้น ถ้าอยู่ในการป้อนภาษาอังกฤษก็ส่งรหัสแอสกีที่อ่านได้ไปแสดงผลบนจอภาพ หากอยู่ในภาวะการป้อนภาษาไทย ก็นำรหัสแอสกีที่ได้ไปเปลี่ยนเป็นภาษาไทยตามตัวแปร ThaiChar$ ก่อนที่จะส่งรหัสแอสกีที่ได้ไปแสดงบนจอภาพ

5. หลังการป้อนข้อความจะได้ตัวแปรสตริง (ตามที่กำหนด) มาหนึ่งค่า ดังนั้นถ้าต้องการตัวเลขก็ให้เปลี่ยนด้วยฟังก์ชัน VAL อีกที

นำแนวคิดข้างต้นนี้มาเขียนชุดคำสั่ง ReadSTR ได้ดังรูปที่ 2 ซึ่งมีรายละเอียดดังนี้

1. บรรทัดที่ 39 กำหนดให้การเรียกใช้ชุดคำสั่งต้องให้ค่า x, y, count, ThaiMode และตัวแปรสตริงเข้ามาด้วย โดยตัวแปรสตริงนี้จะคืนกลับออกมาหลังจบชุดคำสั่งเพื่อนำไปใช้งานต่อไป

2. บรรทัดที่ 40-41 เป็นการกำหนดตำแหน่งของรหัสแอสกีภาษา-

ไทย โดยตัวแรกจะถูกเปลี่ยนมาจากการป้อน ! (รหัสแอสกี 33) ทำให้ตำแหน่งของภาษาไทยที่ตรงกับภาษาอังกฤษคือ รหัสแอสกี-32 นั่นเอง ในการป้อนขอให้ตรวจสอบให้ตรงด้วย

3. บรรทัดที่ 42 กำหนดตัวแปรที่ใช้งาน

4. บรรทัดที่ 43 กำหนดการป้อนจนกว่าจะมีการกดแป้น Enter

5. บรรทัดที่ 44 ส่วนของการแสดงเคอร์เซอร์ โดยตัวแปร Ypix จะทำให้ขนาดของเคอร์เซอร์ในป้อนภาษาไทยและภาษาอังกฤษมีขนาดที่แตกต่างกัน

6. บรรทัดที่ 45-47 วนรอบตรวจการกดแป้น

7. บรรทัดที่ 48 ลบเคอร์เซอร์เพื่อเตรียมแสดงข้อความที่ป้อน

8. บรรทัดที่ 49-55 ส่วนตรวจการกดแป้น ESC เพื่อสลับการป้อนภาษาไทยและภาษาอังกฤษไปมา โดยจะมีการเปลี่ยนค่าตัวแปร ThaiMode และ Ypix

9. บรรทัดที่ 56-62 ถ้ากดแป้นพิมพ์ปกติและตัวแปร C น้อยกว่า Count โดยนำแป้นพิมพ์นั้นมาเปลี่ยนเป็นรหัสแอสกีภาษาไทยในกรณีที่ตัวแปร ThaiMode = 1 จากคำสั่ง A$ = Mid$ (ThaiChr$,ASC(A$) – 32,1) ซึ่งจะได้ตัวแปร A$ ตามรหัสภาษาไทยและภาษาอังกฤษที่ถูกต้องตามการป้อน ไปรวมกับตัวแปร Content$ จากนั้นนำตัวแปร Content$ ที่ได้ไปแสดงบนจอภาพ การที่ผู้เขียนสั่งให้พิมพ์ข้อความใหม่ทุกครั้งเนื่องจากต้องการลดการตรวจการแสดงภาษาไทยให้ถูกต้องในชุดคำสั่งนี้ จึงอาศัยชุดคำสั่ง OutThaiXY ทำงานแทน ถ้าเป็นการป้อนรหัสแอสกีที่มีค่าต่ำกว่า 215 ก็จะเลื่อนเคอร์เซอร์ไปข้างหน้าด้วยการเพิ่มตัวแปร XX

```
39:  SUB ReadSTR(X,Y,Count,ThaiMode,Content$)
40:    Thaichr$="..2347ง6759มขไฝจล/-ภฤิคตชวฒซฟง1ฤ%ฉฏฏไณัษศ?ฯฦOฬฒเธีย"
41:    Thaichr$=ThaiChr$+Chr$(34)+"ๆ(บ\ฐ8'ฬิแก้าดไร่าสทีนยๆพหือไปัผฐ¦,~"
42:    A$=""  :  Xold=X :  XX=X : Content$=""  : C=0  : Ypix=3
43:    WHILE A$<>Chr$(13)
44:        LINE(XX+1,Y+Ypix)-(XX+7,Y+12),3,BF,&HFFFF
45:       DO
46:         A$=Inkey$
47:       LOOP Until A$<>""
48:        LINE(XX+1,Y+Ypix)-(XX+7,Y+12),0,BF,&HFFFF
49:       IF A$=Chr$(27) then
50:           IF ThaiMode=1 then
51:                ThaiMode=0 : Ypix=10
52:              Else
53:                ThaiMode=1 : Ypix=3
54:            END IF
55:       END IF
56:       IF (A$>Chr$(31)) and (C<Count) then
57:          IF (ThaiMode=1) and ((A$>chr$(32) and A$<chr$(127))) then
58:                       A$=Mid$(ThaiChr$,ASC(A$)-32,1)
59:          END IF
60:          Content$=Content$+A$  : CALL OutThaiXY(X,Y,Content$)
61:          C=C+1  : X=Xold       : if A$<Chr$(215) then XX=XX+8
62:       END IF
63:       IF (A$=Chr$(8)) and (C>0) then
64:         IF Right$(Content$,1)<chr$(215) Then
65:              LINE(XX-7,Y-18)-(XX,Y+18),0,BF,&HFFFF
66:              XX=XX-8
67:         END IF
68:         IF Right$(Content$,1)=chr$(215) Then
69:             LINE(XX-7,Y+12)-(XX,Y+18),0,BF,&HFFFF
70:         END IF
71:         IF Right$(Content$,1)=chr$(216) Then
72:               LINE(XX-7,Y+12)-(XX,Y+18),0,BF,&HFFFF
73:         END IF
74:         IF Right$(Content$,1)>chr$(216) Then
75:                LINE(XX-7,Y-16)-(XX,Y-2),0,BF,&HFFFF
76:           IF Mid$(Content$,len(Content$)-1,1)<chr$(217) Then
77:                LINE(XX-7,Y-10)-(XX,Y+2),0,BF,&HFFFF
78:           END IF
79:         END IF
80:         Content$=left$(Content$,Len(Content$)-1)
81:         C=C-1
82:       END IF
83:    WEND
84:  END SUB
```

รูปที่ 2 ชุดคำสั่งการป้อนข้อความกราฟิกด้วยตัวขนาดปกติ

10. บรรทัดที่ 63-82 เป็นการตรวจการกดแป้นลูกศรถอยหลังเพื่อลบอักษรที่ป้อนผิด

11. บรรทัดที่ 64-67 เป็นการลบอักษรในแถวกลาง โดยการวาดรูปสี่เหลี่ยมด้วยสีหมายเลข 0 ในขอบเขตที่กำหนด พร้อมกับเลื่อนค่า XX ลดลงเพื่อเลื่อนเคอร์เซอร์ถอยหลัง คำสั่ง LINE ในบรรทัดที่ 65 คือการลบจอภาพส่วนที่อักษรตัวหลังสุดแสดงอยู่ ค่า &HFFFF คือ การกำหนดรูปแบบพื้นเป็นพื้นทึบทั้งหมด

12. บรรทัดที่ 68-73 เป็นการลบเมื่อตัวหลังสุดเป็น สระ

13. บรรทัดที่ 74-79 เป็นการลบ

เมื่อตัวหลังสุดเป็นสระบนและวรรณยุกต์ โดยจะตรวจสอบว่าอักขระถัดไปอีกหนึ่งตัวเป็นสระบนด้วยหรือไม่ เพื่อลบวรรณยุกต์ที่เขียนซ้อนบนสระบนนั้น ๆ

14. บรรทัดที่ 80-81 ลบตัวอักขระตัวสุดท้ายออกไปจากตัวแปร Content$ และลดค่าตัวแปร C โดยการใช้งานจะเป็นดังนี้

– CALL ReadSTR (10,50, 30,1,Add$) ป้อนค่าให้กับตัวแปร Add$ เริ่มต้นที่ภาษาไทย

– CALL ReadSTR (240,150, 10,0,Num$) ป้อนค่าให้กับตัวแปร Num$ เริ่มต้นที่ภาษาอังกฤษ

ตัวแปร Add$, Tel$ นี้เราสามารถนำไปใช้งานอื่นได้ เช่น No= VAL(Num$) เป็นต้น

อักขระกราฟิกขนาดใหญ่

หากผู้อ่านต้องการแสดงข้อความภาษาไทยและภาษาอังกฤษขนาดใหญ่บนจอภาพด้วยภาษาเบสิกนั้น วิธีการของการกำหนดในตัวแปรเพื่อใช้กับคำสั่ง draw นั้นก็เป็นแนวทางหนึ่งซึ่งผู้เขียนได้นำเสนอในหนังสือที่กล่าวถึงข้างต้นมาบ้างแล้ว

สำหรับบทความนี้ขอเสนอแนวทางใหม่คือ จะใช้ข้อมูลจากไฟล์ข้อมูลสำหรับการพิมพ์มาใช้งาน โดยขอใช้ไฟล์ font1.lq ของโปรแกรมพิมพ์เอกสารราชวิถีเวิร์ดพีซี ซึ่งมีโครงสร้างดังนี้

1. ไฟล์มีความยาว 12,096 ไบต์ มีข้อมูลอยู่ทั้งสิ้น 224 ชุด ดังนั้นข้อมูลจึงเป็น 54 ไบต์ต่อชุด

2. ข้อมูลจำนวน 54 ไบต์ต่อหนึ่งชุดนี้ (หนึ่งตัวอักษร) จะแบ่งออกเป็นชุด ๆ ละ 3 ไบต์จำนวน 18 ชุด

3. ข้อมูลชุด 3 ไบต์นี้จะเรียงจากบนลงล่างไบต์ละ 8 จุด (หนึ่งไบต์มี 8 บิต) ทำให้ได้ข้อมูล 24 จุด (24 บิต) ต่อหนึ่งชุด ซึ่งถ้านำไปลงจุดจะได้จำนวน 24 จุดในแนวตั้ง หนึ่งตัวอักษรจึงสูง 24 จุดกว้าง 18 จุด แต่จะลงจุดเฉพาะส่วนที่บิตนั้น ๆ มีค่าไม่เป็น 0 เท่านั้น จึงทำให้แสดงรูปตัวอักษรตามที่กำหนดได้

หลักการสำคัญจึงอยู่ที่การตรวจสอบว่าบิตใดบ้างที่มีค่าประจำบิตไม่เป็นศูนย์ ซึ่งในการนำข้อมูลไปพิมพ์นี้จะมีค่าประจำบิตเรียงจากบนลงล่างคือ 128, 64,32,16,8,4,2,1 ตามลำดับ โดยอาศัยฟังก์ชัน and มาตรวจสอบ เราก็จะทราบว่าบิตนั้นมีค่าเป็น 0 หรือมีค่าอื่นได้

เมื่อเขียนชุดคำสั่ง OutThaiBig จะเป็นดังรูปที่ 3 ซึ่งมีรายละเอียดดังนี้

```
 85:  SUB OutThaiBig(X,Y,Content$)
 86:   IF FileRead<>27 then
 87:     OPEN "Font1.LQ" AS #2 LEN=12096
 88:     FIELD #2, 12096 AS Font$
 89:     GET #2,1
 90:     CLOSE #2
 91:     ThaiBig$=Font$  : Ccolor=3
 92:   END IF
 93:     Bit(0)=128 : Bit(1)=64  : Bit(2)=32  : Bit(3)=16
 94:     Bit(4)=8   : Bit(5)=4   : Bit(6)=2   : Bit(7)=1
 95:   FileRead=27
 96:   for k=1 to len(Content$)
 97:     Ypix=0  : Xpix=0
 98:    IF Mid$(Content$,k,1)=chr$(3) then
 99:          k=k+1
100:       IF Mid$(Content$,k,1)>chr$(47) and Mid$(Content$,k,1)<chr$(58) then
101:          Ccolor=Asc(Mid$(Content$,k,1))-48 : k=k+1
102:          if Mid$(Content$,k,1)=chr$(1) then Ccolor=Ccolor+9 : k=k+1
103:       END IF
104:    END IF
105:    IF Mid$(Content$,k,1)>chr$(32) then
106:       if Mid$(Content$,k,1)>chr$(214) then Ypix=20 : Xpix=-16
107:         IF Mid$(Content$,k,1)>chr$(216) then
108:            Ypix=-19 : Xpix=-16
109:            if Mid$(Content$,k-1,1)>chr$(216) then Ypix=-28
110:            if Mid$(Content$,k+1,1)=chr$(207) then Ypix=-24
111:         END IF
112:        IF Mid$(Content$,k,1)=chr$(207) then
113:           circle(X-4,Y-1),2,Ccolor
114:           Mid$(Content$,k,1)=chr$(206)
115:        END IF
116:        ThaiPatt$=Mid$(THaiBig$,(ASC(Mid$(Content$,k,1))-32)*54+1,54)
117:        NN=1
118:        for no=1 to 18
119:          for YY=0 to 2
120:            Patt=ASC(Mid$(ThaiPatt$,NN,1)) : NN=NN+1
121:            for B=0 to 7
122:             IF (Patt and Bit(B))=Bit(B) then
123:                Pset(X+Xpix+no,Y+YY*8+Ypix+B),Ccolor
124:             END IF
125:            next
126:          next
127:        next
128:    END IF
129:      if Mid$(Content$,k,1)<chr$(215) then X=X+16
130:   next
131:  END SUB
```

รูปที่ 3 ชุดคำสั่งแสดงข้อความภาษาไทยและภาษาอังกฤษกราฟิกขนาดใหญ่จากไฟล์ font 1. lq

1. บรรทัดที่ 85 ต้องส่งตัวแปร x, y ตำแหน่งเริ่มต้น และข้อความไปให้แสดงบนจอภาพ

2. บรรทัดที่ 86-92 เป็นการอ่านข้อมูลจากไฟล์ font1.lq ซึ่งยาว 12,096 ไบต์ โดยนำค่าที่อ่านได้ใส่ในตัวแปร ThaiBig$ พร้อมกับกำหนดสีเริ่มต้นที่ตัวแปร Ccolor

3. บรรทัดที่ 93-94 กำหนดค่าประจำบิตให้กับตัวแปรมิติ bit เนื่องจากตัวแปรนี้มีจำนวนไม่ถึง 10 จึงไม่ต้องจองหน่วยความจำด้วยคำสั่ง DIM

4. บรรทัดที่ 97 กำหนดตัวแปร Ypix, Xpix ซึ่งช่วยในการแสดงอักขระให้ถูกตำแหน่ง

5. บรรทัดที่ 98-104 เป็นการตรวจสอบการกำหนดสีของอักขระด้วย ^C, ตัวเลข และ ^A

6. บรรทัดที่ 106 กำหนดค่าตัวแปร Xpix, Ypix ใหม่ เมื่อพบว่าเป็นสระล่าง

7. บรรทัดที่ 107-111 กำหนดค่าตัวแปร Xpix, Ypix ใหม่ เมื่อพบว่าเป็นสระบนและวรรณยุกต์ ซึ่งจะตรวจว่าอักขระที่อยู่ก่อนนี้อีกหนึ่งตัว (k-1) เป็นสระบนหรือไม่ และตรวจอักขระข้างหน้า (k+1) ว่าเป็นสระ ำ หรือไม่ ซึ่งจะช่วยให้เขียนวรรณยุกต์ในตำแหน่งที่ต่างกันตามที่ตรวจสอบพบ

8. บรรทัดที่ 112-115 ตรวจสอบสระ ำ เพื่อที่จะให้แสดงข้อความได้สวยงามขึ้น

9. บรรทัดที่ 116 ตัดตัวแปร ThaiBig มาเพียง 54 ไบต์ตามตำแหน่งที่คำนวณได้ มาไว้ในตัวแปรชื่อ ThaiPatt$

10. บรรทัดที่ 117-127 เป็นการนำข้อมูลมาลงจุด โดยการตรวจสอบค่าประจำบิตเพื่อลงจุดให้ตรงกับตำแหน่งที่คำนวณได้จาก Pset(X+Xpix+no, Y+YY*8+Ypix+B),Ccolor ดังนี้

เมื่อ YY=0 จะเริ่มจุดบนสุดที่ตำแหน่ง 0 นับจากจุดบนสุดลงมาข้างล่าง

YY=1 จะเริ่มจุดบนสุดที่ตำแหน่ง 1*8=8 นับจากจุดบนสุดลงมาข้างล่าง

YY=2 จะเริ่มจุดบนสุดที่ตำแหน่ง 2*8=16 นับจากจุดบนสุดลงมาข้างล่าง

ทำให้ข้อมูลแต่ละไบต์ของชุด 3 ไบต์แสดงเรียงจากบนลงล่างจำนวน 24 จุดพอดี ตัวแปร B จะช่วยในการลงจุดระดับบิตอยู่ในตำแหน่งที่ถูกต้องเช่นกัน ตัวแปร NN จะเป็นตัวกำหนดลำดับของข้อมูล และ no จะขยับการลงจุดไปทางขวาทุก ๆ 3 ไบต์

11. บรรทัดที่ 129 กำหนดให้เลื่อนค่า X ไปทางขวาอีก 16 จุด ในกรณีที่ไม่ใช่สระและวรรณยุกต์การใช้งานจะเป็นดังตัวอย่าง

```
CALL OutThaiBig(30, 120,
"การแสดงข้อความภาษาไทย/English")
```

```
1: $INCLUDE "THAI.INC"

2: '================ THAI.BAS  By Arthan Sattayarak ======================
3: screen 9
4:  for k=0 to 6
5:   line(k,k)-(639-k,348-k),k+1,B
6:  next
7:  CALL OutThaiBig(110,20,"ภาษาไทยกราฟิกบนเทอร์โบเบสิก")
8:  CALL OutThaiXY(20,60,"สำหรับการทำงานบนจอสีและจอวีจีเอโมโนโครมโดยเฉพาะ")
9:  CALL OutThaiXY(20,90,"มี 2 ชุดคำสั่ง คือ")
10: CALL OutThaiXY(50,120,"OutThaiXY ตัวขนาดปกติจากไฟล์ THAIENG.FON")
11: CALL OutThaiBig(50,150,"OutThaiBig ตัวขนาดใหญ่จากไฟล์ FONT1.LQ")
12: CALL OutThaiBig(210,190,"ซึ่งเป็นไฟล์ข้อมูลสำหรับเครื่องพิมพ์")
13: CALL OutThaiBig(210,230,"ของราชวิถีเวิร์คพีซี")
14: CALL OutThaiXY(70,265,"มีชุดคำสั่ง ReadSTR สำหรับป้อนข้อความ")
15: CALL OutThaiBig(60,310,"เขียนโดย นายอาจหาญ สัตยารักษ์ ลำปาง")
16: CALL OutThaiXY(190,90,"ป้อนชื่อของท่าน")
17: CALL ReadSTR(290,90,60,1,YourName$)
18: CALL OutThaiXY(350,265,"ชื่อของท่าน "+YourName$)
19: A$=Inkey$
20: END
```

รูปที่ 4 เป็นโปรแกรมหลักทดสอบการแสดงข้อความและการป้อนข้อความ

เมื่อเราได้ชุดคำสั่งมา 3 ชุดแล้ว ให้เก็บโปรแกรมทั้งหมด (รูปที่ 1-3) ไว้ในแผ่นดิสค์ โดยให้ชื่อไฟล์ว่า THAI.INC เพื่อที่ตราวต่อไปเราจะใช้คำสั่ง $INCLUDE "THAI.INC" เรียกเข้ามาใช้งานในกรณีที่ต้องการแสดงข้อความตามแนวคิดนี้

การทดสอบทำได้โดยการเขียนโปรแกรมดังรูปที่ 4 โดยผู้เขียนจะไม่อธิบายในส่วนนี้ เพราะมีเพียง 20 บรรทัดเท่านั้น หวังว่าผู้อ่านคงทำความเข้าใจได้ และเมื่อรันโปรแกรมจะได้ผลบนจอภาพบางส่วนดังรูปที่ 5

การเคลื่อนที่ของวัตถุเมื่อถูกแรงกระทำ

การศึกษาเรื่องการเคลื่อนที่ของวัตถุตามกฎการเคลื่อนที่ของนิวตันในวิชาฟิสิกส์นั้น เราสามารถนำเขียนโปรแกรมช่วยการเรียนการสอนได้ ซึ่งในบทความนี้ขอเสนอโปรแกรมภาษาเบสิก

ให้เป็นตัวอย่างบ้าง โดยมีรายละเอียดที่จะนำไปเขียนโปรแกรมดังนี้

1. วัตถุรูปสี่เหลี่ยมมีมวลขนาด Mass กิโลกรัม วางบนพื้นราบ

2. ออกแรงขนาด Force นิวตันลากวัตถุนี้ให้เคลื่อนที่ โดยให้แรงทำมุม Angle องศากับแนวราบ

3. ในการคำนวณเราต้องแตกแรง Force ออกเป็นแรง 2 แรงที่ตั้งฉากกัน โดยแรง Fx เป็นแรงที่แตกไปตามแนวราบ และแรง Fy เป็นแรงที่แตกไปในแนวดิ่ง โดยคำนวณได้จาก

```
Fx = Force*COS(Angle)
Fy = Force*SIN(Angle)
```

4. หาความเร่งของการเคลื่อนที่ได้จากสูตรการเคลื่อนที่ข้อ 2 ของนิวตัน คือ F = ma จึงได้ความเร่งมีค่า = Fx/Mass เพราะแรงในแนวการเคลื่อนที่มีค่า Fx นิวตัน

5. คำนวณระยะทางที่วัตถุเคลื่อนที่ (เริ่มจากจุดหยุดนิ่ง) ได้จากสมการการเคลื่อนที่ $s = ut + 1/2at^2$ จึงทำให้ระยะทางในแต่ละวินาทีที่ผ่านไปคือ 0.5*a*t*t เพราะว่า ut มีค่าเป็นศูนย์ ระยะทางที่คำนวณได้นำไปแสดงภาพของวัตถุ จะทำให้มองเห็นการเคลื่อนที่ที่มีอัตราเร่ง

ผู้เขียนใช้คำสั่ง DRAW มาช่วยการวาดภาพด้วย ซึ่งมีส่วนกำหนดการเคลื่อนที่ของเส้นที่วาดดังนี้

Un เคลื่อนที่ไปข้างหน้า n จุด

Dn เคลื่อนที่ถอยหลัง n จุด

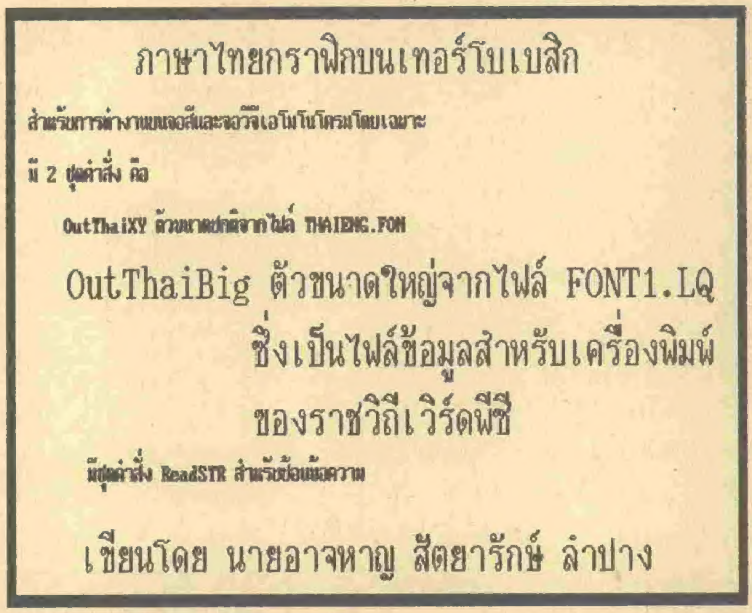
Rn เคลื่อนที่ไปทางขวามือ n จุด

Ln เคลื่อนที่ไปทางซ้ายมือ n จุด

En เคลื่อนที่เฉียงขึ้นทางขวา 45 องศาจำนวน n จุด

Fn เคลื่อนที่เฉียงลงทางขวา 45 องศาจำนวน n จุด

Gn เคลื่อนที่เฉียงลงทางซ้าย 45 องศาจำนวน n จุด



รูปที่ 5 ภาพหน้าจอ ลอกจากการทำงานของโปรแกรมในรูปที่ 4

Hn เคลื่อนที่เฉียงขึ้นทางซ้าย 45 องศาจำนวน n จุด

Cn กำหนดสีเส้นเป็นสีหมายเลข n

TAn กำหนดทิศการเคลื่อนที่เป็นมุม n องศา (จากแนวดิ่ง) n + ไปทางซ้าย และ – ไปทางขวา

Mx,y ไปจากที่เดิม x,y จุดตามแนวราบและแนวดิ่ง

B กำหนดไม่ให้ลากเส้น

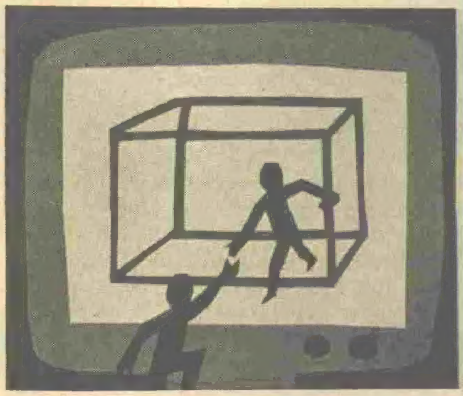
N กำหนดให้ลากเส้นแล้วกลับที่เดิม

เรานำมาเขียนโปรแกรมได้ดังรูปที่ 6 ซึ่งจะเรียกไฟล์ THAI.INC เข้ามาเพื่อใช้แสดงข้อความได้ทั้งภาษาไทยและภาษาอังกฤษ โดยมีรายละเอียดดังนี้

1. บรรทัดที่ 1 เรียกไฟล์ thai.inc เข้ามาใช้งานด้วย

2. บรรทัดที่ 3-6 ลบจอภาพ ตีกรอบสี่เหลี่ยม วาดพื้นราบล้อมข้อความขนาดใหญ่ และวาดพื้นราบ

3. บรรทัดที่ 7-8 กำหนดรูปของวัตถุ โดยจะเริ่มวาดที่มุมบนขวา ลากเส้นลงมา 60 จุด (D60) ลากไปทางซ้าย 60 จุด (L60) ลากขึ้นข้างบน 60 จุด (U60) ลากไปทางขวา 60 จุด (R60) จบด้วยการลากเส้นลงข้างล่าง 30 จุด (D30) ซึ่งเป็นจุดสุดท้ายของการวาดภาพวัตถุ โดยผู้เขียนจะวาดแรงที่กระทำต่อ ณ จุดสิ้นสุดนี้ต่อไป เมื่อกำหนดรูปร่างของวัตถุแล้วก็ทำการวาดรูปของวัตถุโดยให้มุมบนขวาอยู่ที่ตำแหน่ง 70,280



```
1: $INCLUDE "THAI.INC"

2: SUB ForceShow
3:   cls
4:   line(50,0)-(520,50),13,B
5:   CALL OutThaiBig(60,20,"การเคลื่อนที่ของวัตถุเมื่อถูกแรงกระทำ'
6:   line(0,341)-(500,343),4,BF
7:   Box$="D60L60U60R60D30"
8:   Draw  "BM70,280C3"+Box$
9:   CALL OutThaiXy(40,60,"ให้ค่าของแรง        นิวตัน (10-90)"
10:  CALL OutThaiXy(40,90,"ให้มุมที่แรงกระทำกับแนวราบ      องศ
11:  DO
12:   Line(135,55)-(160,85),0,BF,&HFFFF
13:   CALL ReadSTR(135,60,2,0,ST$)
14:   Force=VAL(ST$)
15:  LOOP Until (Force>=10) and (force<=90)
16:  DO
17:   Line(215,85)-(240,115),0,BF,&HFFFF
18:   CALL ReadSTR(215,90,2,0,ST$)
19:   Angle=VAL(ST$)
20:  LOOP Until (Angle>=10) and (angle<=80)
21:    CALL OutThaiXy(40,120,"แตกแรง")
22:    Mass=30 : pi =22/7
23:    Fx=Force*COS((Angle/180)*pi)
24:    Fy=Force*SIN((Angle/180)*pi)
25:    A=Fx/Mass
26:    Fx$=str$(Int(Fx*100)/100)
27:    Fy$=Str$(Int(Fy*100)/100)
28:    A$=Str$(Int(a*100)/100)
29:    CALL OutThaiXy(150,120,"แรงในแนวราบ = "+Fx$+" นิวตัน")
30:    CALL OutThaiXy(150,150,"แรงในแนวดิ่ง  = "+Fy$+" นิวตัน")
31:    CALL OutThaiXy(40,180,"วัตถุจะเคลื่อนที่ด้วยความเร่ง "+A$+" เมตร/วินาที")
32:    CALL OutThaiXy(298+LEN(A$)*8,174,"2")
33:    Angle$="TA"+str$(-(90-Angle))
34:    Force$="U"+str$(Force)+"NG5NF5TA0BR5NR6D4NR4D6"
35:    Draw "BM70,280C3"+Box$+Angle$+Force$
36:   T=0
37:  DO
38:     S=int(0.5*A*T*T)
39:     Draw "TA0BM"+Str$(70+S)+",280C3"+Box$+Angle$+Force$
40:     delay 0.1
41:     Draw "TA0BM"+Str$(70+S)+",280C0"+Box$+Angle$+Force$
42:     T=T+0.4
43:  LOOP Until (70+S)>400
44:   Draw "TA0BM"+Str$(70+S)+",280C2"+Box$+Angle$+Force$
45:  A$=Inkey$
46: END SUB

47: '================= FORCE.BAS =============
48: screen 9
49: CALL ForceShow
```

รูปที่ 6 โปรแกรมการเคลื่อนที่ของวัตถุที่ถูกแรงกระทำ

4. บรรทัดที่ 9-10 แสดงข้อความให้ทราบว่าจะป้อนแรงและมุมของแรงโดยได้กำหนดขอบเขตให้ทราบบนจอภาพด้วย

5. บรรทัดที่ 11-15 ส่วนการป้อนแรง โดยอาศัยชุดคำสั่ง ReadSTR มาช่วย นำค่าที่ป้อนไปเปลี่ยนเป็นค่าของแรง Force โดยจะตรวจสอบว่าป้อนค่าเกินขอบเขตที่กำหนดไว้หรือไม่ ถ้าเกินจะลบส่วนที่ป้อนไว้ด้วยคำสั่งในบรรทัดที่ 12 เสร็จแล้วจึงป้อนค่าใหม่

6. บรรทัดที่ 16-20 ส่วนการป้อนค่ามุมที่กระทำกับแนวราบ โดยนำค่าที่ป้อนนั้นเปลี่ยนเป็นค่า Angle โดยจะตรวจสอบว่าป้อนค่าเกินขอบเขตที่กำหนดไว้หรือไม่

7. บรรทัดที่ 23-25 คำนวณหาค่าของแรงในแนวราบ (Fx) แรงในแนวดิ่ง (Fy) และความเร่ง (A)

8. บรรทัดที่ 26-28 เปลี่ยนค่า Fx, Fy, A มาเป็นข้อความแสดงทศนิยม 2 ตำแหน่ง

9. บรรทัดที่ 29-32 แสดงข้อความให้ทราบค่า Fx, Fy และ A บนจอภาพ

10. บรรทัดที่ 33 คำนวณหามุมในการวาดแรงที่ออก การที่ใช้ 90-Angle เพราะว่า TA นับเทียบกับแนวดิ่ง แต่เรากำหนด Angle เทียบกับแนวราบ และต้องการให้ลากเฉียงไปทางขวาจึงต้องกำหนดด้วยค่ามุมที่เป็น –

11. บรรทัดที่ 34 คำนวณหาภาพของแรงซึ่งจะมีลูกศรที่ปลายแรง (NG5NF5) และเขียนตัวอักษร F ที่ปลายลูกศรด้วย (BR5R6D4NR4D6)

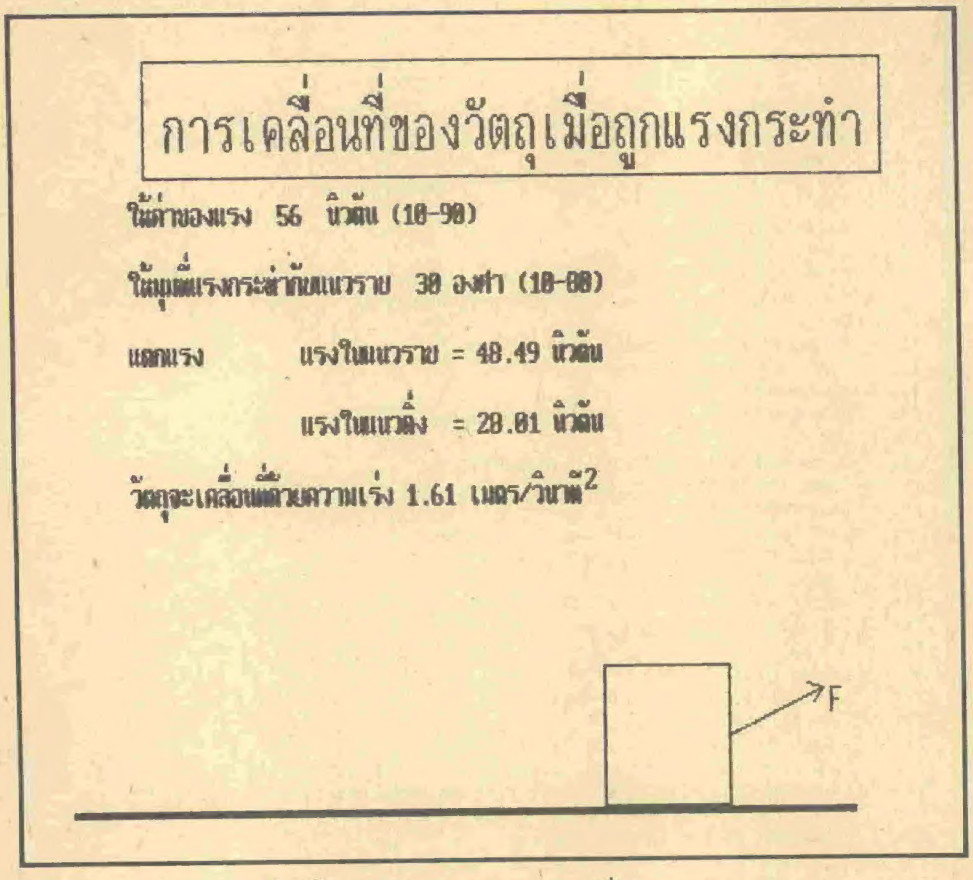
12. บรรทัดที่ 35 ทำการวาดภาพวัตถุ แรงที่กระทำ (ตามทิศทางที่ป้อนไว้) พร้อมตัวอักษร F

13. บรรทัดที่ 36-43 ส่วนการวนรอบแสดงการเคลื่อนที่ของวัตถุ โดยใช้หลักการของการเคลื่อนที่ในแนวราบเมื่อมีแรง Fx กระทำกับวัตถุ โดยเริ่มต้นที่เวลา (T) มีค่าเป็นศูนย์

14. บรรทัดที่ 38 คำนวณหาระยะทางที่วัตถุเคลื่อนที่ได้นับจากจุดเริ่มต้น (จุดเริ่มต้นอยู่ที่ 70)

15. บรรทัดที่ 39 วาดรูปวัตถุ แรง ตัวอักษร F ณ ตำแหน่งที่คำนวณได้ด้วยสีหมายเลข 3

16. บรรทัดที่ 40-41 หลังการหน่วงเวลา เราจะวาดรูปวัตถุ แรง และตัว



รูปที่ 7 ภาพสุดท้ายจากการเคลื่อนที่ของวัตถุจากการรันโปรแกรมในรูปที่ 6

อักษร F ใหม่ ณ ตำแหน่งที่วาดเดิมด้วยสีหมายเลข 0 ซึ่งจะเป็นการลบภาพวัตถุของเดิม

17. บรรทัดที่ 42 เพิ่มค่าเวลาให้มากขึ้น (เปลี่ยนค่า 0.4 ได้ตามความต้องการ)

18. บรรทัดที่ 43 การเคลื่อนที่ของวัตถุจะสิ้นสุดเมื่อวัตถุเคลื่อนที่ไปจนมุมบนขวาอยู่เกินจุด 400 ของแนวราบ

19. บรรทัดที่ 44 วาดรูปวัตถุแรง ตัวอักษร F ที่ตำแหน่งสุดท้ายด้วยสีหมายเลข 3

20. บรรทัดที่ 49 เรียกใช้ชุดคำสั่ง ForceShow เพื่อให้เกิดผลบนจอภาพ

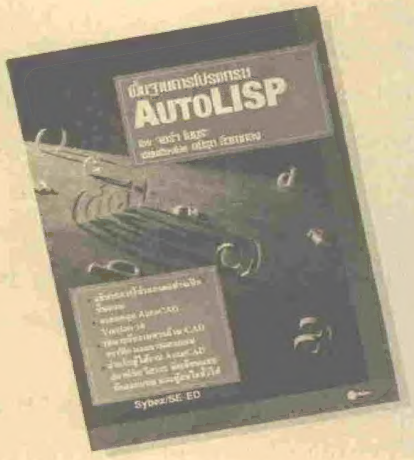
เมื่อลองรันโปรแกรมนี้ ตามด้วยการป้อนค่าแรง และมุมที่ทำกับแนวราบ โปรแกรมจะแสดงข้อความต่าง ๆ ที่เกี่ยวข้องและแสดงการเคลื่อนที่ของวัตถุจากซ้ายไปขวา โดยตัวอย่างหนึ่งของการป้อนจะเกิดผลบนจอภาพดังรูปที่ 7

ก่อนจาก

หวังว่าผู้อ่านที่ชอบภาษาเบสิกคงจะชอบใจที่ได้ชุดคำสั่งแสดงข้อความที่สวยงามถึง 2 ขนาด ทั้งยังได้ชุดคำสั่งการป้อนข้อความนำไปใช้งานได้ ต่อจากนี้ไปโปรแกรมภาษาเบสิกที่ท่านพัฒนาคงช่วยให้ผู้เรียนรู้ไม่ลำบากใจในคำสั่งที่มีแต่ภาษาอังกฤษอีกต่อไป

ซีเอ็ด

สำหรับ
ผู้ใช้ AutoLISP
ที่ไม่เคยเขียนโปรแกรมคอมพิวเตอร์มาก่อน



- ศึกษาวิธีใช้ตัวแปลภาษา AutoLISP
- เทคนิคที่จำเป็นในการออกแบบ และจัดการกับโปรแกรม AutoLISP
- สร้าง เก็บบันทึก และวิธีใช้โปรแกรมตัวอย่างในบทเรียน
- ปรับปรุง เปลี่ยนแปลง และแก้ไขโปรแกรมในบทเรียน ให้มีประสิทธิภาพเพิ่มขึ้น โดยใช้เทคนิคต่าง ๆ มากมาย

พื้นฐานการโปรแกรม
AUTOLISP

หาซื้อได้ตามร้านหนังสือชั้นนำทั่วไป
หรือที่ ซีเอ็ดบุ๊คเซ็นเตอร์ทุกสาขา
และสั่งซื้อได้ที่ซีเอ็ด สำนักงานใหญ่
โทร 248-6280

มีจำหน่ายแล้ว